A Tuning-free Robust and Efficient Approach to High-dimensional Regression

Lan Wang*, Bo Peng, Jelena Bradic*, Runze Li* and Yunan Wu

August 3, 2020 (JSM 2020)

U Miami, Adobe, UC San Diego, Penn State and U Minnesota

Acknowledgments

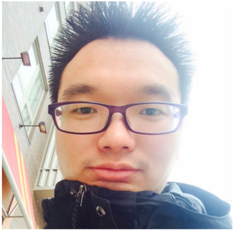
Millions thanks to

Regina Liu and Hongyu Zhao for putting this session together.

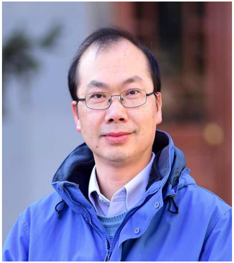
Jianqing Fan, Po-Ling Loh and Ali Shojaie for their inspiring discussions.

Support from NSF grants.

Joint work with



Bo Peng
(Adobe)



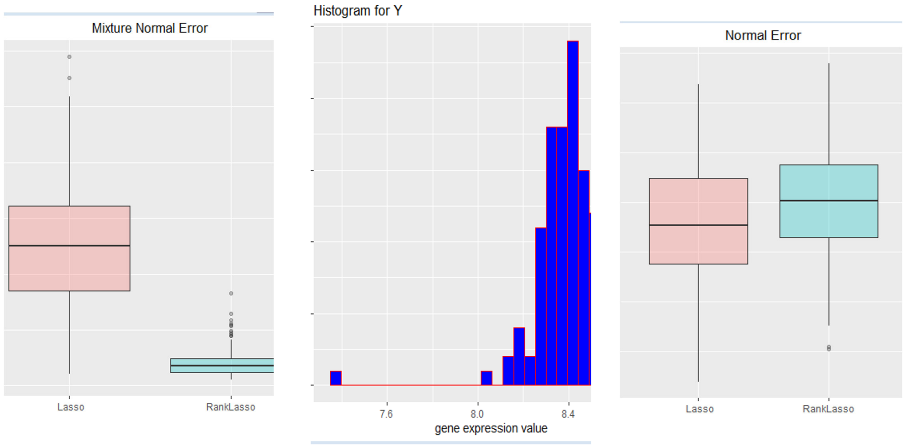
Runze Li
(Penn State)

Jelena Bradic
(UCSD)

Yunan Wu
(UMN)

Introduction

Motivations: large p, tuning, non-normality, efficiency...



High-dimensional regression and Lasso

Suppose that $\{\mathbf{x}_i, y_i\}$, $i = 1, \cdots, n$, is a random sample from a linear regression model

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i,$$

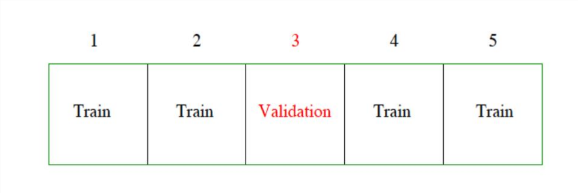
where $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^T$, $\epsilon_1, \ldots, \epsilon_p$ are random errors and $\mathbf{p} \gg \mathbf{n}$.

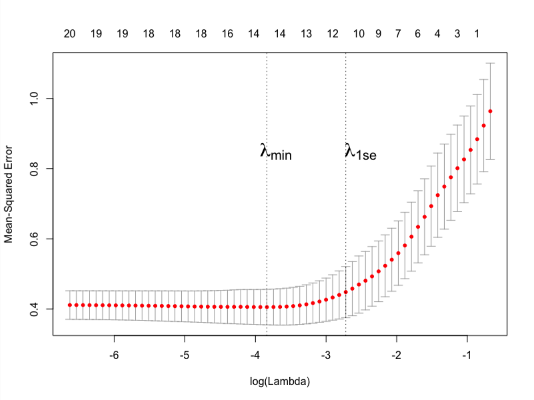
The Lasso is defined to be the minimizer of penalized least squares function:

$$\hat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_1 \right\},$$

where $||\boldsymbol{\beta}||_1 = \sum_{j=1}^{p} |\beta_j|$, $\lambda > 0$ is the tuning parameter.

Cross-validation for Lasso





Gap between the theory and practice of Lasso

- Exiting theory for Lasso is often derived while fixing λ at a **theoretical value**

$$\tau\sigma\sqrt{\log \mathbf{p}/\mathbf{n}},$$

  where σ is the standard deviation of ϵ, and τ is some positive constant.

- Estimating σ in high dimension is itself a very difficult problem.
- Does the cross-validated Lasso share the same near-oracle rate as Lasso does when λ is fixed at the ideal theoretical value?

Gap between the theory and practice of Lasso (cont'd)

- The KKT condition suggests one would choose λ such that

$$P\Big\{||n^{-1}\mathbf{X}^T\boldsymbol{\epsilon}||_\infty \leq \lambda\Big\} \geq 1-\alpha,$$

for some small $\alpha > 0$.

- The optimal choice of λ for Lasso depends on **both the random error distribution and the design matrix**.

Question 1: How to determine the right amount of regularization in a computationally efficient way with proper theoretical justification?

Existing literature

- **Scaled Lasso** (Sun and Zhang, 2012) iteratively estimates the regression parameter and σ (theory under normality).
- **Square-root Lasso** (Belloni et al., 2011) eliminates the need to calibrate λ for σ but does not adjust for the design matrix nor the tail of the random error distribution.

$$\hat{\boldsymbol{\beta}}_{\sqrt{\text{Lasso}}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ n^{-1/2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}.$$

- **TREX** (Lederer and Müller, 2015) automatically adjusts λ for both the tail of the error distribution and the design matrix but the modified loss function is no longer convex.

$$\hat{\boldsymbol{\beta}}_{\text{TREX}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{2\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}{\|\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\|_\infty} + \|\boldsymbol{\beta}\|_1 \right\}.$$

- Sabourin et al. (2015) develops a permutation approach .

Impact of ϵ on tuning parameter and the Lasso solution

Question 2: How to properly handle heavy-tailed error contamination in high dimension so that one achieves robustness while maintaining efficiency for the normal error setting?

- High-dimensional M-estimation based on Huber's loss (Fan et al., 2017; Loh, 2017; Sun et al., 2017): additional tuning parameter.
- Least absolute deviation loss (Belloni et al., 2011; Bradic et al. ,2011; Wang et al., 2012; Wang, 2013; Fan et al. 2014): significant efficiency loss may occur for normal random errors.
- Existing work on high-dimensional robust regression has not addressed the problem of tuning parameter selection and may require some additional tuning parameter to achieve robustness.

Rank-Lasso

To address **Questions** 1 and 2 **simultaneously**, we introduce **Rank-Lasso**.

- Convex and can be computed by linear programming
- Tuning parameter can be easily simulated and automatically adjusts for ϵ and $\mathbf{X}$, with theoretical guarantees
- Almost as efficient as Lasso at normal random errors
- Robust and more efficient than Lasso at heavy-tailed errors

Rank-Lasso

We will consider the following L_1 regularized estimator of $\boldsymbol{\beta}_0$:

$$\widehat{\beta}(\lambda) = \underset{\boldsymbol{\beta}\in\mathcal{R}^p}{\operatorname{argmin}} \Big\{ \big[\mathbf{n}(\mathbf{n}-\mathbf{1})\big]^{-\mathbf{1}} \sum\sum_{\mathbf{i}\neq\mathbf{j}} \big|(\mathbf{Y_i} - \mathbf{x_i^T}\boldsymbol{\beta}) - (\mathbf{Y_j} - \mathbf{x_j^T}\boldsymbol{\beta})\big|$$

$$+ \lambda \sum_{k=1}^{p} |\beta_k| \Big\}.$$

- Minimizing this loss is equivalent to minimizing Jaeckel's dispersion function with Wilcoxon scores (Jaeckel, 1972):

$$\sqrt{12} \sum_{i=1}^{n} \left[\frac{\mathbf{R}(\mathbf{Y_i} - \mathbf{x_i^T}\boldsymbol{\beta})}{n+1} - \frac{1}{2} \right] (Y_i - \mathbf{x}_i^T\boldsymbol{\beta}),$$

where $R(Y_i - \mathbf{x}_i^T\boldsymbol{\beta})$ denotes the rank of $Y_i - \mathbf{x}_i^T\boldsymbol{\beta}$ among $Y_1 - \mathbf{x}_1^T\boldsymbol{\beta}, \ldots, Y_n - \mathbf{x}_n^T\boldsymbol{\beta}$.

Rank-Lasso (cont'd)

With the aid of slack variables $\xi_{ij}^{+}, \xi_{ij}^{-}$, and ζ_k, the convex optimization problem can be equivalently expressed as a **linear programming** problem.

$$
\begin{aligned}
&\min_{\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\zeta}} \Big\{ \big[n(n-1)\big]^{-1} \sum_{i \neq j} \sum (\xi_{ij}^{+} + \xi_{ij}^{-}) + \lambda \sum_{k=1}^{p} \zeta_k \Big\} \\
\text{subject to} \quad & \xi_{ij}^{+} - \xi_{ij}^{-} = (Y_i - Y_j) - (\mathbf{x}_i - \mathbf{x}_j)^T \boldsymbol{\beta}, \;\; i, j = 1, 2, \cdots, n; \\
& \xi_{ij}^{+} \geq 0, \xi_{ij}^{-} \geq 0, \;\; i, j = 1, 2, \cdots, n; \\
& \zeta_k \geq \beta_k, \zeta_k \geq -\beta_k, \;\; k = 1, 2, \cdots, p.
\end{aligned}
$$

Rank-Lasso (cont'd)

- For i.i.d. random errors, $\boldsymbol{\beta}_0$ still bears the interpretation as the effects of the covariates on the conditional mean.

- This is different from other robust loss functions.
  E.g. Huber's loss function:

$$\ell_\alpha(x) = x^2 \mathrm{I}(|x| \leq 1/\alpha) + (\alpha^{-1}|x| - \alpha^{-2})\mathrm{I}(|x| > 1/\alpha)$$

  Requires $\alpha \to 0$ to estimate the mean regression coefficient.

Tuning-free property

Tuning-free property

Completely pivotal property.

- Let $\gamma = \boldsymbol{\beta} - \boldsymbol{\beta}_0$ and write

$$
\begin{aligned}
Q_n(\gamma) &= \left[n(n-1)\right]^{-1} \sum\sum_{i \neq j} \left|(Y_i - \mathbf{x}_i^T \gamma) - (Y_j - \mathbf{x}_j^T \gamma)\right| \\
&= \left[n(n-1)\right]^{-1} \sum\sum_{i \neq j} \left|(\epsilon_i - \epsilon_j) - (\mathbf{x}_i - \mathbf{x}_j)^T \gamma\right|.
\end{aligned}
$$

- Denote the subgradient of $Q_n(\gamma)$ at $\gamma = \mathbf{0}$ (or equivalently $\boldsymbol{\beta} = \boldsymbol{\beta}_0$) by $\mathbf{S}_n = \frac{\partial Q_n(\gamma)}{\partial \boldsymbol{\gamma}}\big|_{\boldsymbol{\gamma}=\mathbf{0}}$. We would like to choose λ such that

$$
P(\lambda > c||\mathbf{S}_n||_\infty) \geq 1 - \alpha_0,
$$

for some constant $c > 1$ and a given small $\alpha_0 > 0$.

Completely pivotal property (cont'd)

We can show that

$$\mathbf{S}_n = \frac{\partial Q_n(\gamma)}{\partial \gamma}\Big|_{\gamma=\mathbf{0}} \quad = \quad -2\big[n(n-1)\big]^{-1} \sum_{j=1}^{n} \mathbf{x}_j \Big(\sum_{\mathbf{i=1, i \neq j}}^{\mathbf{n}} \text{sign}(\epsilon_{\mathbf{j}} - \epsilon_{\mathbf{i}}) \Big).$$

where $\text{sign}(t) = 1$ if $t > 0$, $= -1$ if $t < 0$, and $= 0$ if $t = 0$.

- $$\xi_j = \sum_{i=1, i \neq j}^{n} \text{sign}(\epsilon_j - \epsilon_i) = 2\text{rank}(\epsilon_j) - (n+1),$$

  where $\text{rank}(\epsilon_j)$ is the the rank of ϵ_j among $\{\epsilon_1, \dots, \epsilon_n\}$.

[1] first noted by Parzen et al. (1994) in a different setting

Completely pivotal property (cont'd)

We can show that

$$\mathbf{S}_n = \frac{\partial Q_n(\gamma)}{\partial \gamma}\Big|_{\gamma=\mathbf{0}} \quad = \quad -2\big[n(n-1)\big]^{-1}\sum_{j=1}^{n}\mathbf{x}_j\Big(\sum_{\mathbf{i=1,i\neq j}}^{\mathbf{n}}\mathrm{sign}(\epsilon_{\mathbf{j}}-\epsilon_{\mathbf{i}})\Big).$$

where $\mathrm{sign}(t)=1$ if $t>0$, $=-1$ if $t<0$, and $=0$ if $t=0$.

- $$\xi_j = \sum_{i=1, i\neq j}^{n}\mathrm{sign}(\epsilon_j-\epsilon_i) = 2\mathrm{rank}(\epsilon_j)-(n+1),$$

  where $\mathrm{rank}(\epsilon_j)$ is the the rank of ϵ_j among $\{\epsilon_1,\ldots,\epsilon_n\}$.

- $$(\mathrm{rank}(\epsilon_1),\ldots,\mathrm{rank}(\epsilon_n))^T \sim \mathcal{U}\{1,2,\ldots,n\}^1$$

[1] first noted by Parzen et al. (1994) in a different setting

Completely pivotal property (cont'd)

Lemma
Under the linear model

$$\mathbf{S_n} = -\mathbf{2}\big[\mathbf{n(n-1)}\big]^{-\mathbf{1}}\mathbf{X^T}\boldsymbol{\xi},$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^T$ has a completely known distribution[2] that is independent of the random error distribution.

Given c and α_0, take

$$\lambda^* = \mathbf{c}\ \mathbf{G}^{-\mathbf{1}}_{||\mathbf{S_n}||_\infty}(\mathbf{1} - \alpha_\mathbf{0}),$$

where $G^{-1}_{||\mathbf{S}_n||_\infty}(1 - \alpha_0)$ denotes the $(1 - \alpha_0)$-quantile of the distribution of $||\mathbf{S}_n||_\infty$.

- λ^* does not depend on the distribution of ϵ
- λ^* adapts to the distribution of ϵ and $\mathbf{X}$ **simultaneously**

[2] Simulated by a random permutation of the integers between 1 and n

Completely pivotal property (cont'd)

- **Square-root Lasso** has a partial pivotal property.
  The gradient at $\boldsymbol{\beta}_0$

$$\Big(\sum_{i=1}^{n} \epsilon_i^2\Big)^{1/2} \sum_{i=1}^{n} \mathbf{x}_i \epsilon_i,$$

  does not depend on σ_ϵ but still depends on all other aspects of the distribution of ϵ
- **LAD Lasso** has a complete pivotal property. However, it has significant loss in efficiency if ϵ is normal: often 1.5 times smaller than Rank-Lasso.

Estimation error guarantees

Near-oracle rate of L_2 error bound

Consider the following cone set

$$\Gamma = \left\{\gamma \in \mathbf{R}^p : \|\gamma_{B^c}\|_1 \leq \bar{c}\|\gamma_B\|_1,\ B \subset \{1, 2, \ldots, p\} \text{ and } \|B\|_0 \leq q\right\},$$

where $\bar{c} = \frac{c+1}{c-1}$.

Lemma

(i) Let $\widehat{\gamma}(\lambda) = \widehat{\beta}(\lambda) - \beta_0$. For any $\lambda \geq c\|\mathbf{S}_n\|_\infty$, we have $\widehat{\gamma}(\lambda) \in \Gamma$.

(ii) There exists a universal constant c_0 such that for any positive constant $l > 1$,

$$P(c\|\mathbf{S}_n\|_\infty < lc_0\sqrt{\log p/n}) \geq 1 - 2\exp(-(l^2 - 1)\log p).$$

(iii) If $p > (2/\alpha_0)^{1/3}$, then $\lambda^* < 2c_0\sqrt{\log p/n}$.

Near-oracle rate of L_2 error bound (cont'd)

Recall: $\lambda^* = cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$

Theorem

Suppose (C1)–(C3) hold. If $p > (2/\alpha_0)^{1/3}$, then $\widehat{\boldsymbol{\beta}}(\lambda^*)$ satisfies

$$\left\|\widehat{\boldsymbol{\beta}}(\lambda^*) - \boldsymbol{\beta}_0\right\|_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q \log p}{n}}$$

with probability at least $1 - \alpha_0 - \exp(-2\log p)$.

[3] Fan et al. (2017), Sun et al. (2020)

Near-oracle rate of L_2 error bound (cont'd)

Recall: $\lambda^* = cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$

Theorem

Suppose (C1)–(C3) hold. If $p > (2/\alpha_0)^{1/3}$, then $\widehat{\boldsymbol{\beta}}(\lambda^*)$ satisfies

$$\left\|\widehat{\boldsymbol{\beta}}(\lambda^*) - \boldsymbol{\beta}_0\right\|_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q \log p}{n}}$$

with probability at least $1 - \alpha_0 - \exp(-2\log p)$.

Assumptions on density: f^*, density of $\epsilon_j - \epsilon_i$
$f^*(t) > 0$ for $|t| \leq q\sqrt{\log(p)/n}$

[3] Fan et al. (2017), Sun et al. (2020)

Near-oracle rate of L_2 error bound (cont'd)

Recall: $\lambda^* = cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$

Theorem

Suppose (C1)–(C3) hold. If $p > (2/\alpha_0)^{1/3}$, then $\widehat{\boldsymbol{\beta}}(\lambda^*)$ satisfies

$$\left\|\widehat{\boldsymbol{\beta}}(\lambda^*) - \boldsymbol{\beta}_0\right\|_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q\log p}{n}}$$

with probability at least $1 - \alpha_0 - \exp(-2\log p)$.

Assumptions on density: f^*, density of $\epsilon_j - \epsilon_i$

$f^*(t) > 0$ for $|t| \leq q\sqrt{\log(p)/n}$

Considerably weaker conditions than

[3] Fan et al. (2017), Sun et al. (2020)

Near-oracle rate of L_2 error bound (cont'd)

Recall: $\lambda^* = cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$

Theorem

Suppose (C1)–(C3) hold. If $p > (2/\alpha_0)^{1/3}$, then $\widehat{\beta}(\lambda^*)$ satisfies

$$\left\|\widehat{\beta}(\lambda^*) - \beta_0\right\|_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q\log p}{n}}$$

with probability at least $1-\alpha_0-\exp(-2\log p)$.

Assumptions on density: f^*, density of $\epsilon_j - \epsilon_i$

$$f^*(t) > 0 \text{ for } |t| \leq q\sqrt{\log(p)/n}$$

Considerably weaker conditions than

- Lasso: $\epsilon_i \approx$ sub-Gaussian.

[3] Fan et al. (2017), Sun et al. (2020)

Near-oracle rate of L_2 error bound (cont'd)

Recall: $\lambda^* = cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$

Theorem

Suppose (C1)–(C3) hold. If $p > (2/\alpha_0)^{1/3}$, then $\widehat{\boldsymbol{\beta}}(\lambda^*)$ satisfies

$$\left\|\widehat{\boldsymbol{\beta}}(\lambda^*) - \boldsymbol{\beta}_0\right\|_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q \log p}{n}}$$

with probability at least $1 - \alpha_0 - \exp(-2\log p)$.

Assumptions on density: f^*, density of $\epsilon_j - \epsilon_i$

$$f^*(t) > 0 \text{ for } |t| \leq q\sqrt{\log(p)/n}$$

Considerably weaker conditions than

- Lasso: $\epsilon_i \approx$ sub-Gaussian.
- SQ-root Lasso: $\epsilon_i \approx$ finite variance.

[3] Fan et al. (2017), Sun et al. (2020)

Near-oracle rate of L_2 error bound (cont'd)

Recall: $\lambda^* = cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$

Theorem

Suppose (C1)–(C3) hold. If $p > (2/\alpha_0)^{1/3}$, then $\widehat{\beta}(\lambda^*)$ satisfies

$$\left\|\widehat{\beta}(\lambda^*) - \beta_0\right\|_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q \log p}{n}}$$

with probability at least $1 - \alpha_0 - \exp(-2\log p)$.

Assumptions on density: f^*, density of $\epsilon_j - \epsilon_i$

$$f^*(t) > 0 \text{ for } |t| \leq q\sqrt{\log(p)/n}$$

Considerably weaker conditions than

- Lasso: $\epsilon_i \approx$ sub-Gaussian.
- SQ-root Lasso: $\epsilon_i \approx$ finite variance.
- Huber Lasso: $\epsilon_i \approx$ bounded $1+\delta$ moments.[3]

[3] Fan et al. (2017), Sun et al. (2020)

Bias & Efficiency improvement

Bias reduction and efficiency improvement

A second-stage enhancement with some light tuning to

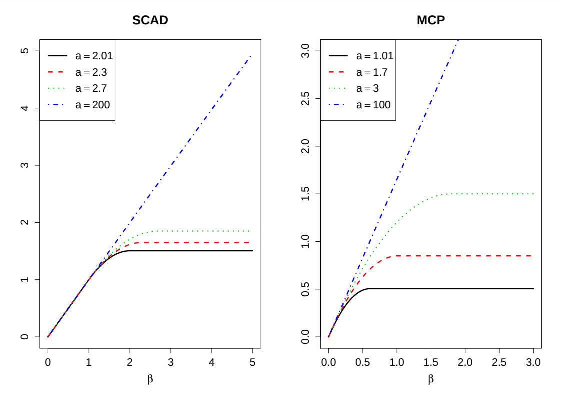
- recover the support of the model with high probability;
- estimate the nonzero coefficients with high efficiency.

Algorithm

$$
\begin{aligned}
\widetilde{\boldsymbol{\beta}}^{(0)} &= \widehat{\boldsymbol{\beta}}(\lambda^*) \text{with simulated tuning parameter} \\
\widetilde{\boldsymbol{\beta}}^{(1)} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T \boldsymbol{\beta})\big| \\
&\qquad + \sum_{k=1}^{p} p'_\eta(|\widetilde{\beta}_k^{(0)}|)|\beta_j| \Big\},
\end{aligned}
$$

p_η being nonconvex penalty with tuning parameter $\eta > 0$

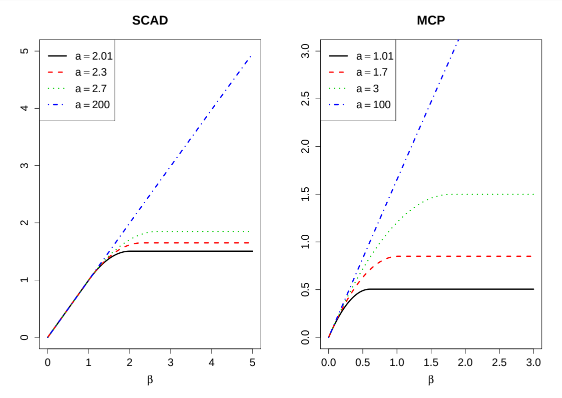
Nonconvex penalties: examples



SCAD penalty(Fan and Li, 2001)

$$
\begin{aligned}
p_\eta(|\beta|) \quad = \quad & \eta|\beta| I(0 \leq |\beta| < \eta) + \frac{a\eta|\beta| - (\beta^2 + \eta^2)/2}{a-1} I(\eta \leq |\beta| \leq a\eta) \\
& + \frac{(a+1)\eta^2}{2} I(|\beta| > a\eta), \quad \text{for some } a > 2.
\end{aligned}
$$

Nonconvex penalties: examples



MCP penalty(Zhang, 2010)

$$p_\eta(|\beta|) = \eta\Big(|\beta| - \frac{\beta^2}{2a\eta}\Big) I(0 \le |\beta| < a\eta) + \frac{a\eta^2}{2} I(|\beta| \ge a\eta), \quad \text{for some } a > 1.$$

Strong oracle property

WLOG, write $\boldsymbol{\beta}_0 = (\boldsymbol{\beta}_{01}^\top, \mathbf{0}_{p-q}^\top)^\top$.

Let

$$\widehat{\boldsymbol{\beta}}_1^{(o)} = \underset{\boldsymbol{\beta}_1}{\operatorname{argmin}} \sum\sum_{i \neq j} \left|(Y_i - \mathbf{x}_{1i}^T \boldsymbol{\beta}_1) - (Y_j - \mathbf{x}_{1j}^T \boldsymbol{\beta}_1)\right|.$$

The oracle estimator for $\boldsymbol{\beta}_0$ is $\widehat{\boldsymbol{\beta}}^o = \left(\widehat{\boldsymbol{\beta}}_1^{(o)\top}, \mathbf{0}_{p-q}^\top\right)^\top$.

Theorem

Under conditions (C1)-(C3), suppose $q = O(n^{c_1})$, $\eta = O(n^{-(1-c_2)/2})$, $\min_{1 \le j \le q} |\beta_{0j}| \ge b n^{-(1-c_3)/2}$, $p = \exp(n^{c_4})$ for some positive constants b and $2c_1 < c_2 < c_3 \le 1$ and $c_1 + c_4 < c_2$, we have

$$P(\widetilde{\boldsymbol{\beta}}^{(1)} = \widehat{\boldsymbol{\beta}}^{(o)}) \to 1, \quad \text{as } n \to \infty.$$

Insights into efficiency

Theorem 2 indicates that whenever q is fixed
$\sqrt{n}(\widetilde{\boldsymbol{\beta}}_1^{(1)} - \boldsymbol{\beta}_{01}) \to \mathcal{N}(0, \sigma^2)$

The relative efficiency (ARE) of $\widetilde{\boldsymbol{\beta}}_1^{(1)}$ with respect to the least-squares oracle for estimating $\boldsymbol{\beta}_{01}$ is

$$\text{ARE} = 12\sigma_\epsilon^2 \left[\int f^2(u)du\right]^2.$$

- $\Rightarrow$ ARE $= 0.955$ for normal error
- $\Rightarrow$ ARE $= 1.5$ for the double exponential distribution,
- $\Rightarrow$ ARE $= 1.9$ for the t_3 distribution.
- $\Rightarrow$ For symmetric error distributions with finite Fisher information, the ARE is known to have a lower bound equal to 0.864.
- $\Rightarrow$ ARE $\approx$ ARE(composite quantile with # of quantiles $\to \infty$)

Light tuning: high-dimensional BIC for second-stage tuning

high-dimensional BIC

$$\mathrm{HBIC}(\eta) = \log\left\{\sum\sum_{\mathbf{i}\neq\mathbf{j}}\left|(\mathbf{Y_i} - \mathbf{x_i^T}\widehat{\boldsymbol{\beta}}_\eta) - (\mathbf{Y_j} - \mathbf{x_j^T}\widehat{\boldsymbol{\beta}}_\eta)\right|\right\} + |\mathbf{A}_\eta|\frac{\log(\log \mathbf{n})}{\mathbf{n}}\mathbf{C_n},$$

where

$$\widehat{\boldsymbol{\beta}}_\eta = \underset{\boldsymbol{\beta}_{A_\eta^C}=0}{\operatorname{argmin}} \sum\sum_{i\neq j}\left|(Y_i - \mathbf{x}_{1i}^T\boldsymbol{\beta}_1) - (Y_j - \mathbf{x}_{1j}^T\boldsymbol{\beta}_1)\right|.$$

$A_\eta = \left\{j : \widetilde{\beta}_{\eta,j}^{(1)} \neq 0, 1 \leq j \leq p\right\}$

$C_n \to \infty,\ (n \to \infty)$: we recommend $C_n = O(\log(p))$.

Consistency of high-dimensional BIC

Find optimal η by

$$\widehat{\eta} = \underset{\eta>0}{\text{argmin}} \ \text{HBIC}(\eta)$$

$$\text{s.t. } |A_\eta| \leq k_n$$

where $k_n > q$ represents a rough estimate of an upper bound of the sparsity size of the underlying model and is allowed to diverge to ∞.

Theorem (Consistency of HBIC)
Assume conditions of Theorem 2 hold, and $k_n \log(p \vee n) = o(\sqrt{n})$.
Assume $\beta^*_{\min} \gg \max\left\{\sqrt{\frac{\log(\log n)}{n}\log p}, \sqrt{\frac{q \log q}{n}}\right\}$.
Then

$$P(A_{\widehat{\eta}} = A) \to 1, \quad \text{as } n \to \infty.$$

where $A = \{j : \beta_{0j} \neq 0, j = 1, \cdots, p\}$.

Numerical examples

Algorithm

Algorithm 1 Incomplete U statistics

1: Compute simulated λ^* given $\mathbf{X}$
2: **for** $i, j \in \{1, \ldots, n\} : i < j$ compute $\mathcal{D} = (\mathbf{x}_i - \mathbf{x}_j, Y_i - Y_j)$
3: Draw a random subsample $\mathcal{S} \subseteq \mathcal{D}$: $|\mathcal{S}| = m$[3]
4: **return** Rank-Lasso

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{m} \sum_{i,j \in \mathcal{S}} \sum_{i \neq j} |Y_i - Y_j - (\mathbf{x}_i - \mathbf{x}_j)^\top \boldsymbol{\beta}| + \lambda \|\boldsymbol{\beta}\|_1$$

Algorithm 2 Simulated λ^*

1: $\boldsymbol{\tau} \leftarrow$ random permutation of $1, \ldots, n$
2: $S \leftarrow 2c\|\mathbf{X}^\top \boldsymbol{\xi}\|_\infty / n(n-1)$, $\boldsymbol{\xi} = 2\tau - (n+1)$
3: Repeat above 100 times and compute $1 - \alpha_0$ quantile of S

[3] Different from sampling m out of n indices and then form pairwise differences

Finite sample: example

We consider: $\mathbf{X} \sim \mathcal{N}(0, \Sigma)$ & $\beta_0 = (\sqrt{3}, \sqrt{3}, \sqrt{3}, 0, ..., 0)$

- $\Sigma = \{\Sigma_1\}_{(i \neq j)} = 0.8$
- $\Sigma = \{\Sigma_1\}_{(i \neq j)} = 0.2$
- $\Sigma = \Sigma_3$ is the AR(1) correlation matrix with coefficient 0.5.

Σ	Error	Method	L1 error	L2 error	ME	FP	FN
Σ_1	$N(0,1)$	Lasso	2.42 (0.06)	0.88 (0.02)	0.17 (0.00)	14.42 (0.46)	0 (0)
		$\sqrt{\text{Lasso}}$	2.22 (0.04)	0.83 (0.01)	0.15 (0.00)	13.09 (0.30)	0 (0)
		Rank Lasso	2.23 (0.04)	0.91 (0.01)	0.25 (0.01)	11.48 (0.28)	0 (0)
		Rank SCAD	0.55 (0.02)	0.34 (0.01)	0.04 (0.00)	0 (0)	0 (0)
	MN	Lasso	4.87 (0.18)	1.75 (0.06)	0.76 (0.04)	12.30 (0.25)	0 (0)
		$\sqrt{\text{Lasso}}$	4.79 (0.17)	1.72 (0.06)	0.75 (0.04)	12.23 (0.26)	0 (0)
		Rank Lasso	0.27 (0.01)	0.11 (0.00)	0.00 (0.00)	12.29 (0.27)	0 (0)
		Rank SCAD	0.05 (0.00)	0.04 (0.00)	0.00 (0.00)	0 (0)	0 (0)
	Cauchy	Lasso	8.70 (0.17)	3.65 (0.05)	5.33 (0.25)	5.54 (0.30)	2.73 (0.04)
		$\sqrt{\text{Lasso}}$	11.14 (0.17)	4.31 (0.09)	4.56 (0.18)	7.85 (0.23)	2.73 (0.54)
		Rank Lasso	5.11 (0.11)	2.03 (0.04)	1.27 (0.04)	11.52 (0.26)	0 (0)
		Rank SCAD	4.07 (0.17)	2.16 (0.08)	1.19 (0.01)	1.12 (0.08)	0.91 (0.05)

Σ	Error	Method	L1 error	L2 error	ME	FP	FN
Σ_2	$N(0,1)$	Lasso	1.21 (0.04)	0.45 (0.01)	0.18 (0.01)	13.08 (0.62)	0 (0)
		$\sqrt{\text{Lasso}}$	0.99 (0.02)	0.44 (0.01)	0.17 (0.00)	6.50 (0.23)	0 (0)
		Rank Lasso	0.94 (0.01)	0.55 (0.01)	0.39 (0.01)	1.34 (0.09)	0 (0)
		Rank SCAD	0.32 (0.01)	0.20 (0.01)	0.04 (0.00)	0.56 (0.07)	0 (0)
	MN	Lasso	2.49 (0.10)	0.92 (0.03)	0.82 (0.05)	11.37 (0.35)	0 (0)
		$\sqrt{\text{Lasso}}$	2.18 (0.08)	0.93 (0.03)	0.81 (0.05)	7.65 (0.23)	0 (0)
		Rank Lasso	0.11 (0.00)	0.07 (0.00)	0.01 (0.00)	1.62 (0.11)	0 (0)
		Rank SCAD	0.03 (0.00)	0.02 (0.00)	0.00 (0.00)	0.40 (0.04)	0 (0)
	Cauchy	Lasso	5.90 (0.10)	2.97 (0.01)	10.00 (0.24)	3.78 (0.32)	2.15 (0.08)
		$\sqrt{\text{Lasso}}$	12.00 (0.37)	3.75 (0.11)	11.19 (0.52)	18.63 (0.33)	1.86 (0.08)
		Rank Lasso	2.36 (0.05)	1.35 (0.03)	2.31 (0.08)	1.45 (0.09)	0 (0)
		Rank SCAD	1.15 (0.05)	0.76 (0.03)	0.56 (0.04)	0.72 (0.06)	0 (0)
Σ_3	$N(0,1)$	Lasso	0.80 (0.03)	0.36 (0.01)	0.14 (0.00)	9.38 (0.60)	0 (0)
		$\sqrt{\text{Lasso}}$	0.71 (0.01)	0.35 (0.01)	0.12 (0.00)	4.47 (0.15)	0 (0)
		Rank Lasso	0.64 (0.01)	0.43 (0.01)	0.25 (0.01)	0 (0)	0 (0)
		Rank SCAD	0.41 (0.02)	0.25 (0.01)	0.04 (0.00)	1.11 (0.13)	0 (0)
	MN	Lasso	1.53 (0.06)	0.67 (0.02)	0.59 (0.04)	7.12 (0.33)	0 (0)
		$\sqrt{\text{Lasso}}$	1.55 (0.06)	0.68 (0.02)	0.57 (0.04)	6.32 (0.19)	0 (0)
		Rank Lasso	0.08 (0.00)	0.05 (0.00)	0.00 (0.00)	0 (0)	0 (0)
		Rank SCAD	0.04 (0.00)	0.02 (0.00)	0.00 (0.00)	0.62 (0.05)	0 (0)
	Cauchy	Lasso	4.96 (0.04)	2.69 (0.04)	11.44 (0.40)	2.57 (0.24)	1.75 (0.09)
		$\sqrt{\text{Lasso}}$	8.99 (0.37)	3.33 (0.12)	11.91 (0.67)	9.78 (0.23)	1.49 (1.17)
		Rank Lasso	1.51 (0.03)	0.99 (0.02)	1.37 (0.05)	0 (0)	0 (0)
		Rank SCAD	1.27 (0.06)	0.82 (0.04)	0.38 (0.03)	0.63 (0.05)	0 (0)

Computation time

Example: $\Sigma_{(i \neq j)} = 0.5$

Error	Method	time/s	L_1 error	L_2 error
$N(0,1)$	Lasso	0.44	0.83 (0.03)	0.28 (0.00)
	$\sqrt{\text{Lasso}}$	8.08	0.70 (0.01)	0.26 (0.00)
	Rank Lasso	0.54	1.07 (0.02)	0.55 (0.01)
	Rank SCAD	3.72	0.47 (0.01)	0.26 (0.01)
Cauchy	Lasso	0.87	7.32 (0.16)	3.16 (0.03)
	$\sqrt{\text{Lasso}}$	10.20	9.31 (0.20)	3.45 (0.06)
	Rank Lasso	0.49	3.84 (0.11)	1.82 (0.05)
	Rank SCAD	3.72	2.86 (0.12)	1.64 (0.07)

A real data example

Goal
Identify genetic variation relevant to hereditary diseases of the retina[4].

Data
y : expression of the gene TRIM32 on 120 male-rats

X : 300 expression quantitative trait locus (eQTL)

- $\Rightarrow$ We conducted 100 random partitions: 60 (training), 60 (testing)
- $\Rightarrow$ We report average (sd.err.) across 100 partitions

Method	L_1 error	L_2 error	model size
Lasso	0.075 (0.001)	0.011 (0.000)	19.50 (1.09)
$\sqrt{\text{Lasso}}$	0.074 (0.001)	0.011 (0.000)	19.09 (0.88)
Rank Lasso	0.080 (0.001)	0.014 (0.001)	6.72 (0.27)
Rank SCAD	0.077 (0.001)	0.012 (0.001)	8.17 (0.39)

[4] Scheetz et al. (2006)

Conclusions

Proposed estimator

- Keeps the convex structure for convenient computation.
- Has a tuning parameter that can be easily simulated and automatically adapts to both the error distribution and the design matrix.
- Is equivariant to scale transformation of the response variable.
- Its L_2 estimation error bound achieves the same near-oracle rate as Lasso does.
- It has similar performance as Lasso does with normal random random error distribution and can be substantially more efficient with heavy-tailed error distribution.
- Its efficiency can be further improved via a second-stage enhancement with some light tuning.

Thank you!

A byproduct: equivariance of the penalized estimator

Lemma

Let $\widehat{\beta}(\lambda^*, \mathbf{Y}, \mathbf{X})$ be the proposed new estimator with the simulated tuning parameter λ^* based on a response vector $\mathbf{Y}$ and a design matrix $\mathbf{X}$. Then

$$\widehat{\beta}(\lambda^*, c\mathbf{Y}, \mathbf{X}) = c\widehat{\beta}(\lambda^*, \mathbf{Y}, \mathbf{X})$$

for any nonzero constant c.

Numerical example: dense setting

Consider the same data generative model as before except that $\boldsymbol{\beta}_0 = (2, 2, 2, 2, 1.75, 1.75, 1.75, 1.5, 1.5, 1.5, 1.25, 1.25, 1.25, 1, 1, 1,$ $0.75, 0.75, 0.75, 0.5, 0.5, 0.5, 0.25, 0.25, 0.25, \mathbf{0}_{p-25})^T$, where $\mathbf{0}_{p-25}$ is a $(p-25)$-dimensional vector of zeros.

Comparing with previous Examples, this is a considerably more challenging scenario with 25 active variables and a number of weak signals.

Numerical example: dense setting

Error	Method	L1 error	L2 error	ME	FP	FN
$N(0,2)$	Lasso	20.31 (0.15)	3.14 (0.02)	5.11 (0.07)	50.78 (0.28)	4.01 (0.10)
	$\sqrt{\text{Lasso}}$	20.64 (0.16)	3.16 (0.02)	5.12 (0.07)	50.9 (0.33)	4.25 (0.10)
	SCAD	15.31 (0.25)	3.12 (0.04)	5.08 (0.13)	8.30 (0.30)	8.32 (0.12)
	Rank Lasso	12.61 (0.12)	2.3 (0.02)	3.37 (0.06)	33.92 (0.35)	0.59 (0.05)
	Rank SCAD	10.11 (0.13)	2.33 (0.03)	2.89 (0.07)	5.74 (0.21)	3.25 (0.08)
MN	Lasso	25.12 (0.47)	3.87 (0.07)	8.11 (0.26)	49.51 (0.37)	6.18 (0.19)
	$\sqrt{\text{Lasso}}$	25.02 (0.44)	3.88 (0.07)	7.93 (0.27)	48.19 (0.49)	6.12 (0.20)
	SCAD	20.40 (0.57)	4.03 (0.10)	8.90 (0.39)	9.05 (0.30)	10.57 (0.23)
	Rank Lasso	5.36 (0.10)	0.97 (0.02)	0.61 (0.02)	36.7 (0.48)	0 (0)
	Rank SCAD	4.62 (0.09)	1.10 (0.02)	0.64 (0.02)	1.15 (0.10)	2.53 (0.05)
$\sqrt{2}t_4$	Lasso	24.76 (0.20)	3.78 (0.03)	7.38 (0.11)	50.82 (0.32)	6.13 (0.11)
	$\sqrt{\text{Lasso}}$	24.56 (0.20)	3.77 (0.03)	7.42 (0.11)	48.68 (0.34)	6.14 (0.11)
	SCAD	20.46 (0.31)	4.07 (0.05)	8.66 (0.21)	8.81 (0.24)	10.76 (0.13)
	Rank Lasso	16.33 (0.17)	2.96 (0.03)	5.58 (0.12)	35.06 (0.36)	1.17 (0.07)
	Rank SCAD	14.38 (0.23)	3.21 (0.05)	5.61 (0.16)	8.17 (0.23)	5.30 (0.15)
Cauchy	Lasso	48.07 (0.53)	8.46 (0.14)	42.38 (1.65)	25.73 (0.98)	19.35 (0.31)
	$\sqrt{\text{Lasso}}$	45.66 (0.46)	8.15 (0.13)	44.69 (1.90)	23.12 (0.87)	19.57 (0.30)
	SCAD	36.53 (0.30)	7.40 (0.08)	249.49 (13.23)	11.19 (0.75)	21.20 (0.31)
	Rank Lasso	30.59 (0.51)	5.33 (0.08)	19.97 (0.65)	35.48 (0.35)	6.81 (0.26)
	Rank SCAD	32.92 (0.57)	6.78 (0.12)	25.7 (0.82)	8.68 (0.27)	13.81 (0.25)